# Aplicaciones de los métodos de reconstrucción estéreo multivista

Dr. Guillermo Gallego

Grupo de Tratamiento de Imágenes
E.T.S.I. Telecomunicación
Universidad Politécnica de Madrid

Seminario VPULab 30/11/2012

# Outline

1. Introduction
   - Motivation
   - Classification of 3D reconstruction methods
2. Image based 3D reconstruction
   - Processing steps
   - SfM. Problem Statement
   - Projective Calibration
   - SfM. Camera resectioning
   - SfM. Initial camera pair estimation
3. Examples
   - VISIRE project (2000)
   - ADREP 3D project (2003). San Lorenzo de El Escorial Monastery
   - ADREP 3D project (2003). Books scene
   - Photo Tourism (2006), PhotoSynth (2008), etc.
   - Multi-View Stereo: PMVS (2007) & CMVS (2010)
   - Autocalibration. Six-Line Conic Variety (SLCV)
   - Video Surfing project. Telefonica R&D (2009)

# Ideas of this talk

1. Advances in technology arise from connections between geometry, algebra, optimization, statistics, etc.
2. So, what's in your toolbox?
   Review two problems: monocular and binocular vision.
3. Overview of recent results in multi-view stereo (3D) reconstruction.

# Ideas of this talk

1. Advances in technology arise from connections between geometry, algebra, optimization, statistics, etc.
2. So, what's in your toolbox?
   Review two problems: monocular and binocular vision.
3. Overview of recent results in multi-view stereo (3D) reconstruction.

# Ideas of this talk

1. Advances in technology arise from connections between geometry, algebra, optimization, statistics, etc.
2. So, what's in your toolbox?
   Review two problems: monocular and binocular vision.
3. Overview of recent results in multi-view stereo (3D) reconstruction.

# Outline

1. Introduction
   - Motivation
   - Classification of 3D reconstruction methods
2. Image based 3D reconstruction
   - Processing steps
   - SfM. Problem Statement
   - Projective Calibration
   - SfM. Camera resectioning
   - SfM. Initial camera pair estimation
3. Examples
   - VISIRE project (2000)
   - ADREP 3D project (2003). San Lorenzo de El Escorial Monastery
   - ADREP 3D project (2003). Books scene
   - Photo Tourism (2006), PhotoSynth (2008), etc.
   - Multi-View Stereo: PMVS (2007) & CMVS (2010)
   - Autocalibration. Six-Line Conic Variety (SLCV)
   - Video Surfing project. Telefonica R&D (2009)

# Motivation

- Understanding human vision has always fascinated people...
  ...and implementing it on a computer for practical applications, too.
- Today's topic: 3D reconstruction.
  Goal: Build a 3D model of the scene under study and find the positions of the objects in front of the camera.
- How to generate such 3D model?
  - Need to study the image formation process (geometry and photometry).
  - Geometric descriptors/primitives at hand: points, planes, lines, conics, quadrics, curves, surfaces, etc.
  - Simplest camera model: pin-hole model (e.g. no lens distortion).

# Motivation

- Understanding human vision has always fascinated people...
  ...and implementing it on a computer for practical applications, too.

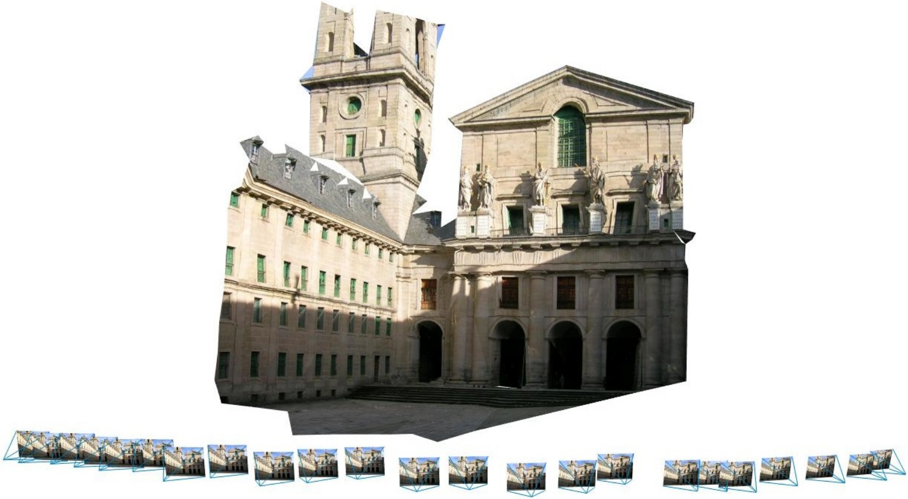
- Today's topic: 3D reconstruction.
  Goal: Build a 3D model of the scene under study and find the positions of the objects in front of the camera.

- How to generate such 3D model?
  - Need to study the image formation process (geometry and photometry).
  - Geometric descriptors/primitives at hand: points, planes, lines, conics, quadrics, curves, surfaces, etc.
  - Simplest camera model: pin-hole model (e.g. no lens distortion).

# Motivation

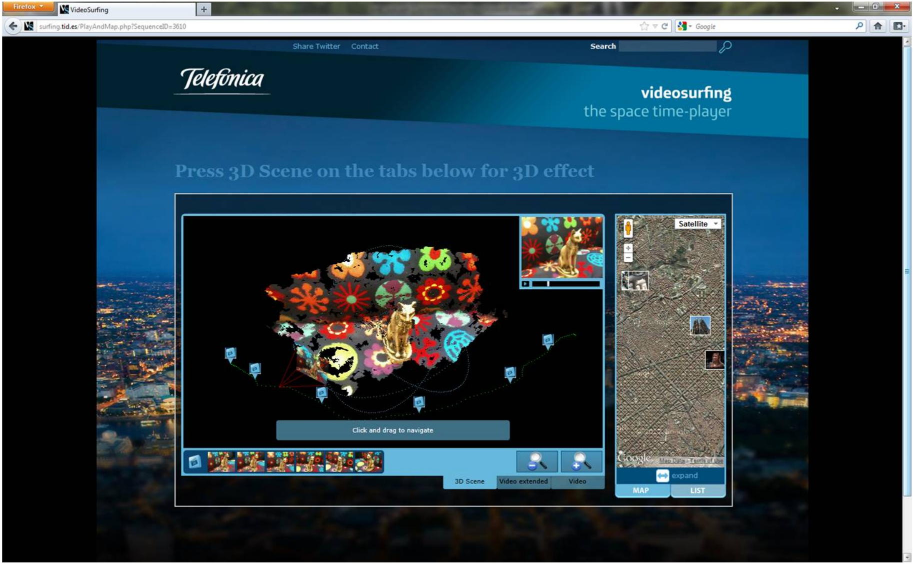
- Understanding human vision has always fascinated people...
  ...and implementing it on a computer for practical applications, too.
- Today's topic: 3D reconstruction.
  Goal: Build a 3D model of the scene under study and find the positions of the objects in front of the camera.
- How to generate such 3D model?
  - Need to study the image formation process (geometry and photometry).
  - Geometric descriptors/primitives at hand: points, planes, lines, conics, quadrics, curves, surfaces, etc.
  - Simplest camera model: pin-hole model (e.g. no lens distortion).

# 3D reconstruction examples



# 3D reconstruction examples: Video Surfing

http://vimeo.com/15990190



# Outline

1. Introduction
   - Motivation
   - Classification of 3D reconstruction methods
2. Image based 3D reconstruction
   - Processing steps
   - SfM. Problem Statement
   - Projective Calibration
   - SfM. Camera resectioning
   - SfM. Initial camera pair estimation
3. Examples
   - VISIRE project (2000)
   - ADREP 3D project (2003). San Lorenzo de El Escorial Monastery
   - ADREP 3D project (2003). Books scene
   - Photo Tourism (2006), PhotoSynth (2008), etc.
   - Multi-View Stereo: PMVS (2007) & CMVS (2010)
   - Autocalibration. Six-Line Conic Variety (SLCV)
   - Video Surfing project. Telefonica R&D (2009)

# Classification of 3D reconstruction methods

- Pre-calibrated (camera parameters are known)

| | |
|---|---|
| Image based | Bottom-up approach: from local image features to building complex 3D models. |
| Voxel based | Space discretization. No full 3D model of scene objects. |
| Object based | Top-down approach: relating 3D models to image features. |

- Online Calibrated: camera parameters are estimated using...

| | |
|---|---|
| Scene constraints | Calibration methods (right angles of the objects, calibration pattern, etc.) |
| Geometric constraints | Autocalibration methods. |

# Classification of 3D reconstruction methods

- Pre-calibrated (camera parameters are known)

| | |
|---|---|
| Image based | Bottom-up approach: from local image features to building complex 3D models. |
| Voxel based | Space discretization. No full 3D model of scene objects. |
| Object based | Top-down approach: relating 3D models to image features. |

- Online Calibrated: camera parameters are estimated using...

| | |
|---|---|
| Scene constraints | Calibration methods (right angles of the objects, calibration pattern, etc.) |
| Geometric constraints | Autocalibration methods. |

# Classification of 3D reconstruction methods

- Pre-calibrated (camera parameters are known)

| | |
|---|---|
| Image based | Bottom-up approach: from local image features to building complex 3D models. |
| Voxel based | Space discretization. No full 3D model of scene objects. |
| Object based | Top-down approach: relating 3D models to image features. |

- Online Calibrated: camera parameters are estimated using...

| | |
|---|---|
| Scene constraints | Calibration methods (right angles of the objects, calibration pattern, etc.) |
| Geometric constraints | Autocalibration methods. |

# Outline

# Processing Steps

1. Digital image acquisition
2. Feature detection and extraction
3. Structure from Motion
    1. Projective calibration
    2. Auto-calibration
    3. Euclidean reconstruction
4. 3D triangulation and texture selection
5. VRML presentation

# Processing Steps

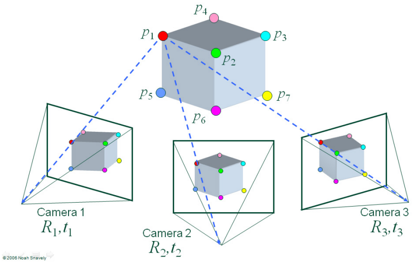
1. Digital image acquisition
2. Feature detection and extraction
3. Structure from Motion
    1. Projective calibration
    2. Auto-calibration
    3. Euclidean reconstruction
4. 3D triangulation and texture selection
5. VRML presentation

# Outline

1. Introduction
   - Motivation
   - Classification of 3D reconstruction methods
2. Image based 3D reconstruction
   - Processing steps
   - SfM. Problem Statement
   - Projective Calibration
   - SfM. Camera resectioning
   - SfM. Initial camera pair estimation
3. Examples
   - VISIRE project (2000)
   - ADREP 3D project (2003). San Lorenzo de El Escorial Monastery
   - ADREP 3D project (2003). Books scene
   - Photo Tourism (2006), PhotoSynth (2008), etc.
   - Multi-View Stereo: PMVS (2007) & CMVS (2010)
   - Autocalibration. Six-Line Conic Variety (SLCV)
   - Video Surfing project. Telefonica R&D (2009)

- Projective geometry in the plane $\mathbb{P}^2$ and in space $\mathbb{P}^3$ (homogeneous coordinates).
- Pin-hole camera model: $\mathbf{x} \sim \mathtt{P}\mathbf{X}$.
- Projective camera model $\mathtt{P} = (\pi_1, \pi_2, \pi_3)^\top$. Axial planes $\{\pi_1, \pi_2\}$, principal plane $\pi_3$.
- Finite (or Euclidean) camera matrix: $\mathtt{P} \sim \mathtt{K}[\mathtt{R} \,|\, \mathbf{t}]$ (in Ch. 2, my $\mathtt{K} \equiv \mathtt{A}$).
- Epipolar geometry: Fundamental matrix $\mathtt{F}$. Epipolar constraint $\mathbf{x}'^\top \mathbf{F}\mathbf{x} = 0$.
- Some linear algebra (QR decomposition, SVD, linear solvers, etc.)

# Structure from motion. Problem Statement.




Given the projections of $n$ points in $m$ images

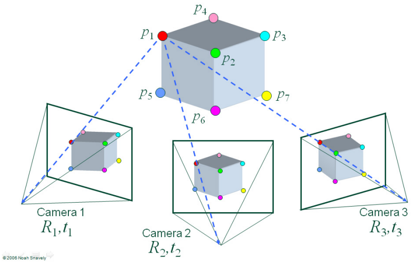
$$\{\mathbf{x}_j^i\}_{i=1,\ldots,m;\, j=1,\ldots,n},$$

compute the 3D points $\{\mathbf{X}_j\}_{j=1,\ldots,n}$
and the camera parameters $\{\mathrm{K}^i, \mathrm{R}^i, \widetilde{\mathbf{C}}^i\}_{i=1,\ldots,m}$
such that the projections agree with the measurements:

$$\mathbf{x}_j^i \sim \mathrm{K}^i[\mathrm{R}^i \mid -\mathrm{R}^i\widetilde{\mathbf{C}}^i]\mathbf{X}_j$$

Typically, # equations $= 2mn \gg$ # unknowns $= 11m + 3n - 7$.

# Structure from motion. Problem Statement.



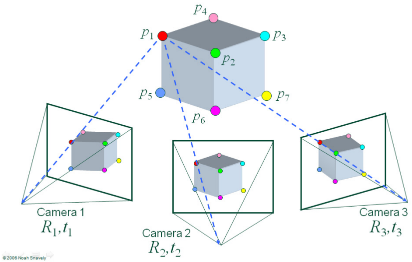

Given the projections of $n$ points in $m$ images

$$\{\mathbf{x}^i_j\}_{i=1,\dots,m;\, j=1,\dots,n},$$

compute the 3D points $\{\mathbf{X}_j\}_{j=1,\dots,n}$
and the camera parameters $\{\mathtt{K}^i, \mathtt{R}^i, \widetilde{\mathbf{C}}^i\}_{i=1,\dots,m}$
such that the projections agree with the measurements:

$$\mathbf{x}^i_j \sim \mathtt{K}^i[\mathtt{R}^i \mid -\mathtt{R}^i\widetilde{\mathbf{C}}^i]\mathbf{X}_j$$

Typically, # equations $= 2mn \gg$ # unknowns $= 11m + 3n - 7$.




Given the projections of $n$ points in $m$ images

$$\{\mathbf{x}_j^i\}_{i=1,\ldots,m;\, j=1,\ldots,n},$$

compute the 3D points $\{\mathbf{X}_j\}_{j=1,\ldots,n}$
and the camera parameters $\{\mathtt{K}^i, \mathtt{R}^i, \widetilde{\mathbf{C}}^i\}_{i=1,\ldots,m}$
such that the projections agree with the measurements:

$$\mathbf{x}_j^i \sim \mathtt{K}^i[\mathtt{R}^i \mid -\mathtt{R}^i\widetilde{\mathbf{C}}^i]\mathbf{X}_j$$

Typically, $\#$ equations $= 2mn \gg \#$ unknowns $= 11m + 3n - 7$.

# Structure from motion. Problem Statement.

## Optimization approach

In the presence of Gaussian noise on the coordinates of the observed points in the images, seek the Maximum Likelihood solution, which is the one that minimizes the reprojection error

$$\sum_{i,j} d^2_{\mathsf{Euc}}\left(\mathbf{x}^i_j, \mathrm{K}^i[\mathrm{R}^i| - \mathrm{R}^i\widetilde{\mathbf{C}}^i]\mathbf{X}_j\right).$$

## Solution: Bundle Adjustment

Solve the above optimization problem in a large number of variables (and equations).

### Optimization approach

In the presence of Gaussian noise on the coordinates of the observed points in the images, seek the Maximum Likelihood solution, which is the one that minimizes the reprojection error

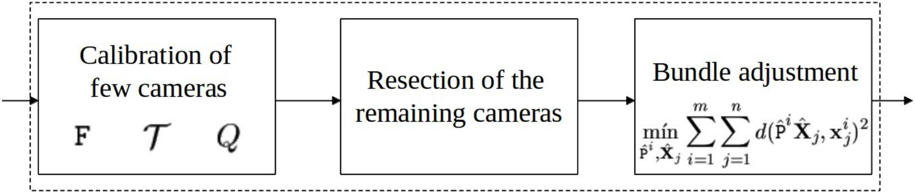
$$\sum_{i,j} d^2_{\mathsf{Euc}}\left(\mathbf{x}^i_j, \mathrm{K}^i[\mathrm{R}^i \,|\, -\mathrm{R}^i\widetilde{\mathbf{C}}^i]\mathbf{X}_j\right).$$

### Solution: Bundle Adjustment

Solve the above optimization problem in a large number of variables (and equations).

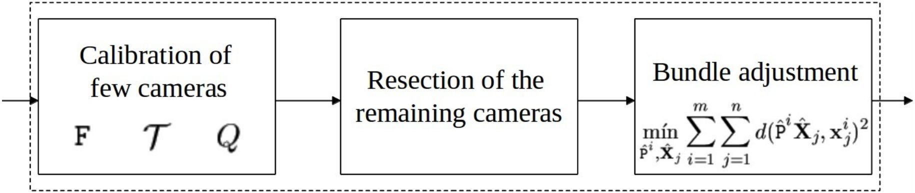
# Outline

# SfM. Projective Calibration

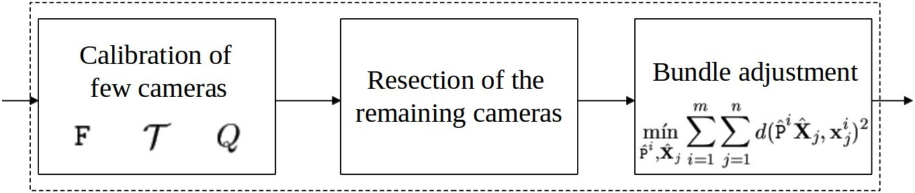



- Unique reconstruction is obtained up to a projective transformation (homography) of space, $\mathbf{x}_j^i \sim \mathrm{P}^i \mathbf{X}_j = (\mathrm{P}^i \mathrm{H})(\mathrm{H}^{-1} \mathbf{X}_j) = \hat{\mathrm{P}}^i \hat{\mathbf{X}}_j$.
- Geometric entities based on the number of cameras:
  - 2 cameras: Fundamental matrix $\mathrm{F}$
  - 3 cameras: Trifocal tensor $\mathscr{T}$
  - 4 camera: Quadrifocal tensor $\mathcal{Q}$
  - Multi-camera geometry: bundle adjustment

# SfM. Projective Calibration



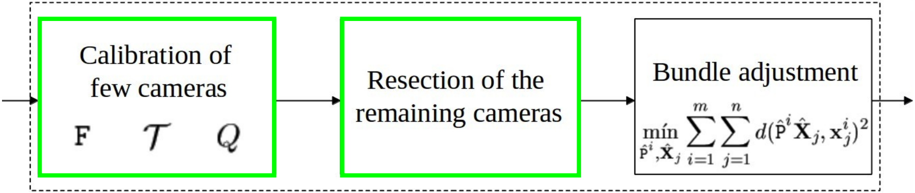

- Unique reconstruction is obtained up to a projective transformation (homography) of space, $\mathbf{x}_j^i \sim \mathrm{P}^i \mathbf{X}_j = (\mathrm{P}^i \mathrm{H})(\mathrm{H}^{-1} \mathbf{X}_j) = \hat{\mathrm{P}}^i \hat{\mathbf{X}}_j$.
- Geometric entities based on the number of cameras:
  - 2 cameras: Fundamental matrix $\mathrm{F}$
  - 3 cameras: Trifocal tensor $\mathscr{T}$
  - 4 camera: Quadrifocal tensor $\mathcal{Q}$
  - Multi-camera geometry: bundle adjustment




- Unique reconstruction is obtained up to a projective transformation (homography) of space, $\mathbf{x}_j^i \sim \mathrm{P}^i \mathbf{X}_j = (\mathrm{P}^i \mathrm{H})(\mathrm{H}^{-1} \mathbf{X}_j) = \hat{\mathrm{P}}^i \hat{\mathbf{X}}_j$.

- Geometric entities based on the number of cameras:
  - 2 cameras: Fundamental matrix $\mathrm{F}$
  - 3 cameras: Trifocal tensor $\mathscr{T}$
  - 4 camera: Quadrifocal tensor $Q$
  - Multi-camera geometry: bundle adjustment

# SfM. Projective Calibration




- Unique reconstruction is obtained up to a projective transformation (homography) of space, $\mathbf{x}_j^i \sim \mathtt{P}^i \mathbf{X}_j = (\mathtt{P}^i \mathtt{H})(\mathtt{H}^{-1} \mathbf{X}_j) = \hat{\mathtt{P}}^i \hat{\mathbf{X}}_j$.
- Geometric entities based on the number of cameras:
  - 2 cameras: Fundamental matrix $\mathtt{F}$
  - 3 cameras: Trifocal tensor $\mathscr{T}$
  - 4 camera: Quadrifocal tensor $\mathcal{Q}$
  - Multi-camera geometry: bundle adjustment

# Outline

Useful for building up incremental reconstructions (3D models) adding one camera to an existing 3D model.

## Problem statement

Given a set of $n$ point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$, compute the projection matrix $\mathrm{P}$ of the camera such that $\mathbf{x}_i \sim \mathrm{P}\mathbf{X}_i$ for all $i = 1, \ldots, n$.

Estimation methods:

- Linear algorithm (algebraic cost, SVD, eigenvalues)
- Non-linear algorithm (geometric cost, Gauss-Newton method)
- Robust algorithm (RANSAC)

Useful for building up incremental reconstructions (3D models) adding one camera to an existing 3D model.

### Problem statement

Given a set of $n$ point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$, compute the projection matrix P of the camera such that $\mathbf{x}_i \sim \mathrm{P}\mathbf{X}_i$ for all $i = 1, \dots, n$.

Estimation methods:

- Linear algorithm (algebraic cost, SVD, eigenvalues)
- Non-linear algorithm (geometric cost, Gauss-Newton method)
- Robust algorithm (RANSAC)

- Projection equation: $\mathbf{x}_i \sim \mathtt{P}\mathbf{X}_i$.
  Data: $\{\mathbf{x}_i \leftrightarrow \mathbf{X}_i\}_{i=1}^{n}$. Unknown: P.

- How can we measure whether $\mathbf{x}_i \sim \mathtt{P}\mathbf{X}_i$ is satisfied or not?

- Cross product: $\mathbf{x}_i \times \mathtt{P}\mathbf{X}_i \stackrel{?}{=} \mathbf{0}$. This is a set of 3 linear equations in the entries of P. If $\mathbf{x}_i = (x_i, y_i, w_i)^\top$,

$$\underbrace{\begin{pmatrix} \mathbf{0}^\top & -w_i\mathbf{X}_i^\top & y_i\mathbf{X}_i^\top \\ w_i\mathbf{X}_i^\top & \mathbf{0}^\top & -x_i\mathbf{X}_i^\top \\ -y_i & x_i\mathbf{X}_i^\top & \mathbf{0}^\top \end{pmatrix}}_{\mathtt{A}_i\,(3\times 12)} \underbrace{\begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{pmatrix}}_{\mathbf{p}} = \mathbf{0}.$$

  But only 2 of them are linearly independent.

- Projection equation: $\mathbf{x}_i \sim \mathtt{P}\mathbf{X}_i$.
  Data: $\{\mathbf{x}_i \leftrightarrow \mathbf{X}_i\}_{i=1}^{n}$. Unknown: P.
- How can we measure whether $\mathbf{x}_i \sim \mathtt{P}\mathbf{X}_i$ is satisfied or not?
- Cross product: $\mathbf{x}_i \times \mathtt{P}\mathbf{X}_i \overset{?}{=} \mathbf{0}$. This is a set of 3 linear equations in the entries of P. If $\mathbf{x}_i = (x_i, y_i, w_i)^\top$,

$$\underbrace{\begin{pmatrix} \mathbf{0}^\top & -w_i\mathbf{X}_i^\top & y_i\mathbf{X}_i^\top \\ w_i\mathbf{X}_i^\top & \mathbf{0}^\top & -x_i\mathbf{X}_i^\top \\ -y_i & x_i\mathbf{X}_i^\top & \mathbf{0}^\top \end{pmatrix}}_{\mathtt{A}_i\,(3\times 12)} \underbrace{\begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{pmatrix}}_{\mathbf{p}} = \mathbf{0}.$$

  But only 2 of them are linearly independent.

- Projection equation: $\mathbf{x}_i \sim \mathtt{P}\mathbf{X}_i$.
  Data: $\{\mathbf{x}_i \leftrightarrow \mathbf{X}_i\}_{i=1}^{n}$. Unknown: $\mathtt{P}$.
- How can we measure whether $\mathbf{x}_i \sim \mathtt{P}\mathbf{X}_i$ is satisfied or not?
- Cross product: $\mathbf{x}_i \times \mathtt{P}\mathbf{X}_i \overset{?}{=} \mathbf{0}$. This is a set of 3 linear equations in the entries of $\mathtt{P}$. If $\mathbf{x}_i = (x_i, y_i, w_i)^\top$,

$$\underbrace{\begin{pmatrix} \mathbf{0}^\top & -w_i\mathbf{X}_i^\top & y_i\mathbf{X}_i^\top \\ w_i\mathbf{X}_i^\top & \mathbf{0}^\top & -x_i\mathbf{X}_i^\top \\ -y_i & x_i\mathbf{X}_i^\top & \mathbf{0}^\top \end{pmatrix}}_{\mathtt{A}_i\,(3\times 12)} \underbrace{\begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{pmatrix}}_{\mathbf{p}} = \mathbf{0}.$$

But only 2 of them are linearly independent.

How do you know there are only 2 independent equations?

Keep using your linear algebra toolbox: Kronecker and vec operator (stack a matrix column-wise).

$$\mathrm{vec}(\mathtt{ABC}) = (\mathtt{C}^\top \otimes \mathtt{A})\,\mathrm{vec}(\mathtt{B}).$$

- Let $\varepsilon_i \doteq \mathbf{x}_i \times \mathtt{P}\mathbf{X}_i = [\mathbf{x}_i]_\times \mathtt{P}\mathbf{X}_i \in \mathbb{R}^3$.
- Since it is a vector,
$\varepsilon_i = \mathrm{vec}(\varepsilon_i^\top) = \mathrm{vec}\big(([\mathbf{x}_i]_\times \mathtt{P}\mathbf{X}_i)^\top\big) = ([\mathbf{x}_i]_\times^\top \otimes \mathbf{X}_i^\top)\,\mathrm{vec}(\mathtt{P}) = -\mathtt{A}_i\mathbf{p}$.
- Now, since $\mathrm{rank}(\mathtt{A}\otimes\mathtt{B}) = \mathrm{rank}(\mathtt{A})\,\mathrm{rank}(\mathtt{B})$, we see that

$$\mathrm{rank}([\mathbf{x}_i]_\times^\top \otimes \mathbf{X}_i^\top) = \mathrm{rank}([\mathbf{x}_i]_\times^\top)\,\mathrm{rank}(\mathbf{X}_i^\top) = 2\cdot 1 = 2.$$

How do you know there are only 2 independent equations?

Keep using your linear algebra toolbox: Kronecker and vec operator (stack a matrix column-wise).

$$\mathrm{vec}(\mathtt{ABC}) = (\mathtt{C}^\top \otimes \mathtt{A})\,\mathrm{vec}(\mathtt{B}).$$

- Let $\varepsilon_i \doteq \mathbf{x}_i \times \mathrm{P}\mathbf{X}_i = [\mathbf{x}_i]_\times \mathrm{P}\mathbf{X}_i \in \mathbb{R}^3$.
- Since it is a vector,
  $\varepsilon_i = \mathrm{vec}(\varepsilon_i^\top) = \mathrm{vec}\big(([\mathbf{x}_i]_\times \mathrm{P}\mathbf{X}_i)^\top\big) = ([\mathbf{x}_i]_\times^\top \otimes \mathbf{X}_i^\top)\,\mathrm{vec}(\mathrm{P}) = -\mathtt{A}_i\mathbf{p}.$
- Now, since $\mathrm{rank}(\mathtt{A} \otimes \mathtt{B}) = \mathrm{rank}(\mathtt{A})\,\mathrm{rank}(\mathtt{B})$, we see that

$$\mathrm{rank}([\mathbf{x}_i]_\times^\top \otimes \mathbf{X}_i^\top) = \mathrm{rank}([\mathbf{x}_i]_\times^\top)\,\mathrm{rank}(\mathbf{X}_i^\top) = 2 \cdot 1 = 2.$$

# SfM. Camera resectioning

## Linear algorithm for the estimation of P

How do you know there are only 2 independent equations?

Keep using your linear algebra toolbox: Kronecker and vec operator (stack a matrix column-wise).

$$\mathrm{vec}(\mathtt{ABC}) = (\mathtt{C}^\top \otimes \mathtt{A})\,\mathrm{vec}(\mathtt{B}).$$

- Let $\varepsilon_i \doteq \mathbf{x}_i \times \mathrm{P}\mathbf{X}_i = [\mathbf{x}_i]_\times \mathrm{P}\mathbf{X}_i \in \mathbb{R}^3$.
- Since it is a vector,
  $\varepsilon_i = \mathrm{vec}(\varepsilon_i^\top) = \mathrm{vec}\big(([\mathbf{x}_i]_\times \mathrm{P}\mathbf{X}_i)^\top\big) = ([\mathbf{x}_i]_\times^\top \otimes \mathbf{X}_i^\top)\,\mathrm{vec}(\mathtt{P}) = -\mathtt{A}_i\mathbf{p}.$
- Now, since $\mathrm{rank}(\mathtt{A}\otimes\mathtt{B}) = \mathrm{rank}(\mathtt{A})\,\mathrm{rank}(\mathtt{B})$, we see that

$$\mathrm{rank}([\mathbf{x}_i]_\times^\top \otimes \mathbf{X}_i^\top) = \mathrm{rank}([\mathbf{x}_i]_\times^\top)\,\mathrm{rank}(\mathbf{X}_i^\top) = 2 \cdot 1 = 2.$$

How do you know there are only 2 independent equations?

Keep using your linear algebra toolbox: Kronecker and vec operator (stack a matrix column-wise).

$$\mathrm{vec}(\mathtt{ABC}) = (\mathtt{C}^\top \otimes \mathtt{A})\,\mathrm{vec}(\mathtt{B}).$$

- Let $\varepsilon_i \doteq \mathbf{x}_i \times \mathtt{P}\mathbf{X}_i = [\mathbf{x}_i]_\times \mathtt{P}\mathbf{X}_i \in \mathbb{R}^3$.
- Since it is a vector, $\varepsilon_i = \mathrm{vec}(\varepsilon_i^\top) = \mathrm{vec}\big(([\mathbf{x}_i]_\times \mathtt{P}\mathbf{X}_i)^\top\big) = ([\mathbf{x}_i]_\times^\top \otimes \mathbf{X}_i^\top)\,\mathrm{vec}(\mathtt{P}) = -\mathtt{A}_i\mathbf{p}$.
- Now, since $\mathrm{rank}(\mathtt{A} \otimes \mathtt{B}) = \mathrm{rank}(\mathtt{A})\,\mathrm{rank}(\mathtt{B})$, we see that

$$\mathrm{rank}([\mathbf{x}_i]_\times^\top \otimes \mathbf{X}_i^\top) = \mathrm{rank}([\mathbf{x}_i]_\times^\top)\,\mathrm{rank}(\mathbf{X}_i^\top) = 2 \cdot 1 = 2.$$

How do you know there are only 2 independent equations?

Keep using your linear algebra toolbox: Kronecker and vec operator (stack a matrix column-wise).

$$\operatorname{vec}(\mathtt{ABC}) = (\mathtt{C}^\top \otimes \mathtt{A}) \operatorname{vec}(\mathtt{B}).$$

- Let $\varepsilon_i \doteq \mathbf{x}_i \times \mathrm{P}\mathbf{X}_i = [\mathbf{x}_i]_\times \mathrm{P}\mathbf{X}_i \in \mathbb{R}^3$.
- Since it is a vector,
  $\varepsilon_i = \operatorname{vec}(\varepsilon_i^\top) = \operatorname{vec}\big(([\mathbf{x}_i]_\times \mathrm{P}\mathbf{X}_i)^\top\big) = ([\mathbf{x}_i]_\times^\top \otimes \mathbf{X}_i^\top) \operatorname{vec}(\mathtt{P}) = -\mathtt{A}_i\mathbf{p}$.
- Now, since $\operatorname{rank}(\mathtt{A} \otimes \mathtt{B}) = \operatorname{rank}(\mathtt{A}) \operatorname{rank}(\mathtt{B})$, we see that

$$\operatorname{rank}([\mathbf{x}_i]_\times^\top \otimes \mathbf{X}_i^\top) = \operatorname{rank}([\mathbf{x}_i]_\times^\top) \operatorname{rank}(\mathbf{X}_i^\top) = 2 \cdot 1 = 2.$$

# SfM. Camera resectioning

## Linear algorithm for the estimation of $\mathtt{P}$

- Having 2 independent equations $\mathtt{A}_i\mathbf{p} = \mathbf{0}$ per $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$ correspondence. How many do we need to estimate $\mathbf{p} \equiv \mathtt{P}$?
- Staking equations, we get a homogeneous linear system of equations

$$\mathtt{A}\mathbf{p} = \begin{pmatrix} \mathtt{A}_1 \\ \mathtt{A}_2 \\ \vdots \\ \mathtt{A}_n \end{pmatrix} \mathbf{p} = \mathbf{0}.$$

- Some more linear algebra...
  - Trivial solution $\mathbf{p} = \mathbf{0}$ (makes no sense as a projection matrix).
  - Exact solution $\mathbf{p} \sim \ker(\mathtt{A})$ is unique (up to a scale) if $\operatorname{rank}(\mathtt{A}) = 11$.
  - Assuming points in general position, we need $2n \geq 11$ equations. Therefore, $n \geq 6$ point correspondences.

- Having 2 independent equations $\mathtt{A}_i\mathbf{p} = \mathbf{0}$ per $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$ correspondence. How many do we need to estimate $\mathbf{p} \equiv \mathtt{P}$?
- Staking equations, we get a homogeneous linear system of equations

$$\mathtt{A}\mathbf{p} = \begin{pmatrix} \mathtt{A}_1 \\ \mathtt{A}_2 \\ \vdots \\ \mathtt{A}_n \end{pmatrix} \mathbf{p} = \mathbf{0}.$$

- Some more linear algebra...
  - Trivial solution $\mathbf{p} = \mathbf{0}$ (makes no sense as a projection matrix).
  - Exact solution $\mathbf{p} \sim \ker(\mathtt{A})$ is unique (up to a scale) if $\operatorname{rank}(\mathtt{A}) = 11$.
  - Assuming points in general position, we need $2n \geq 11$ equations. Therefore, $n \geq 6$ point correspondences.

- Having 2 independent equations $\mathtt{A}_i\mathbf{p} = \mathbf{0}$ per $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$ correspondence. How many do we need to estimate $\mathbf{p} \equiv \mathtt{P}$?
- Staking equations, we get a homogeneous linear system of equations

$$\mathtt{A}\mathbf{p} = \begin{pmatrix} \mathtt{A}_1 \\ \mathtt{A}_2 \\ \vdots \\ \mathtt{A}_n \end{pmatrix} \mathbf{p} = \mathbf{0}.$$

- Some more linear algebra...
  - Trivial solution $\mathbf{p} = \mathbf{0}$ (makes no sense as a projection matrix).
  - Exact solution $\mathbf{p} \sim \ker(\mathtt{A})$ is unique (up to a scale) if $\operatorname{rank}(\mathtt{A}) = 11$.
  - Assuming points in general position, we need $2n \geq 11$ equations. Therefore, $n \geq 6$ point correspondences.

- Having 2 independent equations $\mathtt{A}_i\mathbf{p} = \mathbf{0}$ per $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$ correspondence. How many do we need to estimate $\mathbf{p} \equiv \mathtt{P}$?
- Staking equations, we get a homogeneous linear system of equations

$$\mathtt{A}\mathbf{p} = \begin{pmatrix} \mathtt{A}_1 \\ \mathtt{A}_2 \\ \vdots \\ \mathtt{A}_n \end{pmatrix} \mathbf{p} = \mathbf{0}.$$

- Some more linear algebra...
  - Trivial solution $\mathbf{p} = \mathbf{0}$ (makes no sense as a projection matrix).
  - Exact solution $\mathbf{p} \sim \ker(\mathtt{A})$ is unique (up to a scale) if $\operatorname{rank}(\mathtt{A}) = 11$.
  - Assuming points in general position, we need $2n \geq 11$ equations. Therefore, $n \geq 6$ point correspondences.

- Having 2 independent equations $\mathtt{A}_i\mathbf{p} = \mathbf{0}$ per $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$ correspondence. How many do we need to estimate $\mathbf{p} \equiv \mathtt{P}$?

- Staking equations, we get a homogeneous linear system of equations

$$\mathtt{A}\mathbf{p} = \begin{pmatrix} \mathtt{A}_1 \\ \mathtt{A}_2 \\ \vdots \\ \mathtt{A}_n \end{pmatrix} \mathbf{p} = \mathbf{0}.$$

- Some more linear algebra...
  - Trivial solution $\mathbf{p} = \mathbf{0}$ (makes no sense as a projection matrix).
  - Exact solution $\mathbf{p} \sim \ker(\mathtt{A})$ is unique (up to a scale) if $\operatorname{rank}(\mathtt{A}) = 11$.
  - Assuming points in general position, we need $2n \geq 11$ equations. Therefore, $n \geq 6$ point correspondences.

- What if correspondences are not exact, i.e., there is noise?
  - An exact solution may not exist (case $\text{rank}(\mathtt{A}) = 12$).
  - What do we do?

- **Optimization** approach.
  Find the best $\mathbf{p} \equiv \mathtt{P}$ in the 2-norm sense (least-squares) by minimizing the residual,

$$\min_{\mathbf{p}} \|\mathtt{A}\mathbf{p}\|. \quad (1)$$

- Issue: $\mathbf{p} = \mathbf{0}$ still solves (1).

- Fix: to have a well-posed problem (and to get rid of the scale ambiguity), supplement (1) with an additional constraint on $\mathbf{p}$. Typically, choose $\|\mathbf{p}\| = 1$. Why?

- What if correspondences are not exact, i.e., there is noise?
  - An exact solution may not exist (case $\mathrm{rank}(\mathtt{A}) = 12$).
  - What do we do?

- Optimization approach.
  Find the best $\mathbf{p} \equiv \mathtt{P}$ in the 2-norm sense (least-squares) by minimizing the residual,
  $$\min_{\mathbf{p}} \|\mathtt{A}\mathbf{p}\|. \quad (1)$$

- Issue: $\mathbf{p} = \mathbf{0}$ still solves (1).

- Fix: to have a well-posed problem (and to get rid of the scale ambiguity), supplement (1) with an additional constraint on $\mathbf{p}$. Typically, choose $\|\mathbf{p}\| = 1$. Why?

- What if correspondences are not exact, i.e., there is noise?
  - An exact solution may not exist (case $\mathrm{rank}(\mathtt{A}) = 12$).
  - What do we do?

- **Optimization** approach.
  Find the best $\mathbf{p} \equiv \mathtt{P}$ in the 2-norm sense (least-squares) by minimizing the residual,

$$\min_{\mathbf{p}} \|\mathtt{A}\mathbf{p}\|. \tag{1}$$

- Issue: $\mathbf{p} = \mathbf{0}$ still solves (1).

- Fix: to have a well-posed problem (and to get rid of the scale ambiguity), supplement (1) with an additional constraint on $\mathbf{p}$. Typically, choose $\|\mathbf{p}\| = 1$. Why?

- What if correspondences are not exact, i.e., there is noise?
  - An exact solution may not exist (case $\mathrm{rank}(\mathtt{A}) = 12$).
  - What do we do?

- **Optimization** approach.
  Find the best $\mathbf{p} \equiv \mathtt{P}$ in the 2-norm sense (least-squares) by minimizing the residual,
  $$\min_{\mathbf{p}} \|\mathtt{A}\mathbf{p}\|. \tag{1}$$

- Issue: $\mathbf{p} = \mathbf{0}$ still solves (1).

- Fix: to have a well-posed problem (and to get rid of the scale ambiguity), supplement (1) with an additional constraint on $\mathbf{p}$. Typically, choose $\|\mathbf{p}\| = 1$. Why?

- The resulting problem is a classic: minimization of a Rayleigh quotient,

$$\hat{\mathbf{p}} = \arg \min_{\|\mathbf{p}\|=1} \|\mathtt{A}\mathbf{p}\| = \arg\min_{\mathbf{p}} \frac{\|\mathtt{A}\mathbf{p}\|^2}{\|\mathbf{p}\|^2} = \arg\min_{\mathbf{p}} \frac{\mathbf{p}^\top \mathtt{A}^\top \mathtt{A}\mathbf{p}}{\mathbf{p}^\top \mathbf{p}}. \quad (2)$$

- Solution: Using Lagrange multipliers, compute the extremals of $F(\mathbf{p},\lambda) = \|\mathtt{A}\mathbf{p}\|^2 + \lambda(1 - \|\mathbf{p}\|^2)$.
  - $\hat{\mathbf{p}}$ is the eigenvector of $\mathtt{A}^\top\mathtt{A}$ corresponding to the smallest eigenvalue: $(\mathtt{A}^\top\mathtt{A})\hat{\mathbf{p}} = \lambda_{\min}\hat{\mathbf{p}}$.
  - Using the SVD: $\hat{\mathbf{p}}$ is the right singular vector of $\mathtt{A}$ corresponding to the smallest singular value.
- But, what is being minimized?
  An algebraic cost of the verification of the projection equations.
- Can we choose a (better) *geometrically meaningful* cost to minimize?

- The resulting problem is a classic: minimization of a Rayleigh quotient,

$$\hat{\mathbf{p}} = \arg \min_{\|\mathbf{p}\|=1} \|\mathtt{A}\mathbf{p}\| = \arg\min_{\mathbf{p}} \frac{\|\mathtt{A}\mathbf{p}\|^2}{\|\mathbf{p}\|^2} = \arg\min_{\mathbf{p}} \frac{\mathbf{p}^\top \mathtt{A}^\top \mathtt{A}\mathbf{p}}{\mathbf{p}^\top \mathbf{p}}. \tag{2}$$

- Solution: Using Lagrange multipliers, compute the extremals of $F(\mathbf{p},\lambda) = \|\mathtt{A}\mathbf{p}\|^2 + \lambda(1 - \|\mathbf{p}\|^2)$.
  - $\hat{\mathbf{p}}$ is the eigenvector of $\mathtt{A}^\top\mathtt{A}$ corresponding to the smallest eigenvalue: $(\mathtt{A}^\top\mathtt{A})\hat{\mathbf{p}} = \lambda_{\min}\hat{\mathbf{p}}$.
  - Using the SVD: $\hat{\mathbf{p}}$ is the right singular vector of $\mathtt{A}$ corresponding to the smallest singular value.
- But, what is being minimized?
  An algebraic cost of the verification of the projection equations.
- Can we choose a (better) *geometrically meaningful* cost to minimize?

- The resulting problem is a classic: minimization of a Rayleigh quotient,

$$\hat{\mathbf{p}} = \arg \min_{\|\mathbf{p}\|=1} \|\mathtt{A}\mathbf{p}\| = \arg\min_{\mathbf{p}} \frac{\|\mathtt{A}\mathbf{p}\|^2}{\|\mathbf{p}\|^2} = \arg\min_{\mathbf{p}} \frac{\mathbf{p}^\top \mathtt{A}^\top \mathtt{A}\mathbf{p}}{\mathbf{p}^\top \mathbf{p}}. \quad (2)$$

- Solution: Using Lagrange multipliers, compute the extremals of $F(\mathbf{p},\lambda) = \|\mathtt{A}\mathbf{p}\|^2 + \lambda(1 - \|\mathbf{p}\|^2)$.
  - $\hat{\mathbf{p}}$ is the eigenvector of $\mathtt{A}^\top\mathtt{A}$ corresponding to the smallest eigenvalue: $(\mathtt{A}^\top\mathtt{A})\hat{\mathbf{p}} = \lambda_{\min}\hat{\mathbf{p}}$.
  - Using the SVD: $\hat{\mathbf{p}}$ is the right singular vector of $\mathtt{A}$ corresponding to the smallest singular value.

- But, what is being minimized?
  An algebraic cost of the verification of the projection equations.

- Can we choose a (better) *geometrically meaningful* cost to minimize?

- The resulting problem is a classic: minimization of a Rayleigh quotient,

$$\hat{\mathbf{p}} = \arg \min_{\|\mathbf{p}\|=1} \|\mathtt{A}\mathbf{p}\| = \arg\min_{\mathbf{p}} \frac{\|\mathtt{A}\mathbf{p}\|^2}{\|\mathbf{p}\|^2} = \arg\min_{\mathbf{p}} \frac{\mathbf{p}^\top \mathtt{A}^\top \mathtt{A}\mathbf{p}}{\mathbf{p}^\top \mathbf{p}}. \qquad (2)$$

- Solution: Using Lagrange multipliers, compute the extremals of $F(\mathbf{p}, \lambda) = \|\mathtt{A}\mathbf{p}\|^2 + \lambda(1 - \|\mathbf{p}\|^2)$.
  - $\hat{\mathbf{p}}$ is the eigenvector of $\mathtt{A}^\top\mathtt{A}$ corresponding to the smallest eigenvalue: $(\mathtt{A}^\top\mathtt{A})\hat{\mathbf{p}} = \lambda_{\min}\hat{\mathbf{p}}$.
  - Using the SVD: $\hat{\mathbf{p}}$ is the right singular vector of $\mathtt{A}$ corresponding to the smallest singular value.

- But, what is being minimized?
  An algebraic cost of the verification of the projection equations.

- Can we choose a (better) *geometrically meaningful* cost to minimize?

- Can we choose a (better) *geometrically meaningful* cost to minimize?
- Yes! Since there is a metric in the image plane, minimize the Euclidean distance

$$g(\mathbf{p}) = \sum_{i=1}^{n} d^2_{\mathsf{Euc}}(\mathbf{x}_i, \mathrm{P}\mathbf{X}_i).$$

- Again, a geometry + optimization framework!
- Solution: use standard techniques from finite-dimensional optimization (your toolbox).
  - Steepest descent method
  - Conjugate gradient method
  - Newton's method
- Why didn't we start here? Solution method is slower and iterative.

- Can we choose a (better) *geometrically meaningful* cost to minimize?
- Yes! Since there is a metric in the image plane, minimize the Euclidean distance

$$g(\mathbf{p}) = \sum_{i=1}^{n} d^2_{\mathsf{Euc}}(\mathbf{x}_i, \mathrm{P}\mathbf{X}_i).$$

- Again, a geometry + optimization framework!

- Solution: use standard techniques from finite-dimensional optimization (your toolbox).
  - Steepest descent method
  - Conjugate gradient method
  - Newton's method
- Why didn't we start here? Solution method is slower and iterative.

- Can we choose a (better) *geometrically meaningful* cost to minimize?
- Yes! Since there is a metric in the image plane, minimize the Euclidean distance

$$g(\mathbf{p}) = \sum_{i=1}^{n} d^2_{\text{Euc}}(\mathbf{x}_i, \mathrm{P}\mathbf{X}_i).$$

- Again, a geometry + optimization framework!
- Solution: use standard techniques from finite-dimensional optimization (your toolbox).
  - Steepest descent method
  - Conjugate gradient method
  - Newton's method
- Why didn't we start here? Solution method is slower and iterative.

- Can we choose a (better) *geometrically meaningful* cost to minimize?
- Yes! Since there is a metric in the image plane, minimize the Euclidean distance

$$g(\mathbf{p}) = \sum_{i=1}^{n} d^2_{\mathsf{Euc}}(\mathbf{x}_i, \mathrm{P}\mathbf{X}_i).$$

- Again, a geometry + optimization framework!
- Solution: use standard techniques from finite-dimensional optimization (your toolbox).
  - Steepest descent method
  - Conjugate gradient method
  - Newton's method
- Why didn't we start here? Solution method is slower and iterative.

- Newton's method.
  Assume a quadratic approximation of the cost function

$$g(\mathbf{p}+\delta\mathbf{p}) \approx g(\mathbf{p}) + \nabla g \cdot \delta\mathbf{p} + \tfrac{1}{2}\delta\mathbf{p}^\top H\,\delta\mathbf{p}, \qquad (3)$$

  where $\nabla g$ and $H$ are the gradient and the Hessian of $g$ evaluated at $\mathbf{p}$.

- Iteration: start from $\mathbf{p}^0$ and update $\mathbf{p}^{k+1} = \mathbf{p}^k + (\delta\mathbf{p})^k$ using step

$$H(\mathbf{p}^k)\,(\delta\mathbf{p})^k = -\nabla g(\mathbf{p}^k).$$

- What initialization $\mathbf{p}^0$? For example, the linear method.

- Newton's method.
  Assume a quadratic approximation of the cost function

$$g(\mathbf{p}+\delta\mathbf{p}) \approx g(\mathbf{p}) + \nabla g \cdot \delta\mathbf{p} + \tfrac{1}{2}\delta\mathbf{p}^\top H\,\delta\mathbf{p}, \tag{3}$$

  where $\nabla g$ and $H$ are the gradient and the Hessian of $g$ evaluated at $\mathbf{p}$.

- Iteration: start from $\mathbf{p}^0$ and update $\mathbf{p}^{k+1} = \mathbf{p}^k + (\delta\mathbf{p})^k$ using step

$$H(\mathbf{p}^k)\,(\delta\mathbf{p})^k = -\nabla g(\mathbf{p}^k).$$

- What initialization $\mathbf{p}^0$? For example, the linear method.

- Newton's method.
  Assume a quadratic approximation of the cost function

$$g(\mathbf{p}+\delta\mathbf{p}) \approx g(\mathbf{p}) + \nabla g \cdot \delta\mathbf{p} + \tfrac{1}{2}\delta\mathbf{p}^\top H\,\delta\mathbf{p}, \tag{3}$$

  where $\nabla g$ and $H$ are the gradient and the Hessian of $g$ evaluated at $\mathbf{p}$.

- Iteration: start from $\mathbf{p}^0$ and update $\mathbf{p}^{k+1} = \mathbf{p}^k + (\delta\mathbf{p})^k$ using step

$$H(\mathbf{p}^k)\,(\delta\mathbf{p})^k = -\nabla g(\mathbf{p}^k).$$

- What initialization $\mathbf{p}^0$? For example, the linear method.

Use RANSAC [Fischler & Bolles 1987], etc. to remove data outliers.

- Compute a candidate $\mathtt{P}$
  - Select a random minimal sample (6 correspondences) and compute $\mathtt{P}$ (linear algorithm).
  - Calculate the "distance" from each correspondence to the model.
  - Compute the number of inliers consistent with the model $\mathtt{P}$.
- Choose the $\mathtt{P}$ with the largest number of inliers.
- Refine $\mathtt{P}$ by minimizing the geometric cost (non-linear algorithm) using only the inliers.

Use RANSAC [Fischler & Bolles 1987], etc. to remove data outliers.

- Compute a candidate P
  - Select a random minimal sample (6 correspondences) and compute P (linear algorithm).
  - Calculate the "distance" from each correspondence to the model.
  - Compute the number of inliers consistent with the model P.
- Choose the P with the largest number of inliers.
- Refine P by minimizing the geometric cost (non-linear algorithm) using only the inliers.

Use RANSAC [Fischler & Bolles 1987], etc. to remove data outliers.

- Compute a candidate P
  - Select a random minimal sample (6 correspondences) and compute P (linear algorithm).
  - Calculate the "distance" from each correspondence to the model.
  - Compute the number of inliers consistent with the model P.
- Choose the P with the largest number of inliers.
- Refine P by minimizing the geometric cost (non-linear algorithm) using only the inliers.

Use RANSAC [Fischler & Bolles 1987], etc. to remove data outliers.

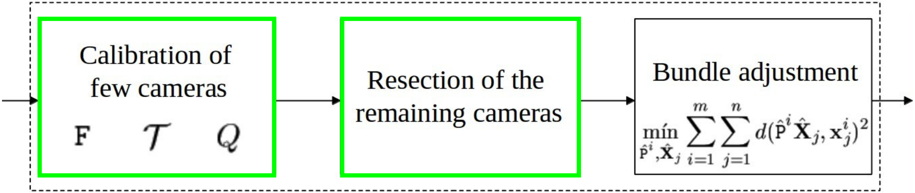
- Compute a candidate P
  - Select a random minimal sample (6 correspondences) and compute P (linear algorithm).
  - Calculate the "distance" from each correspondence to the model.
  - Compute the number of inliers consistent with the model P.
- Choose the P with the largest number of inliers.
- Refine P by minimizing the geometric cost (non-linear algorithm) using only the inliers.

# Outline

# SfM. Projective Calibration

Recall where we are:

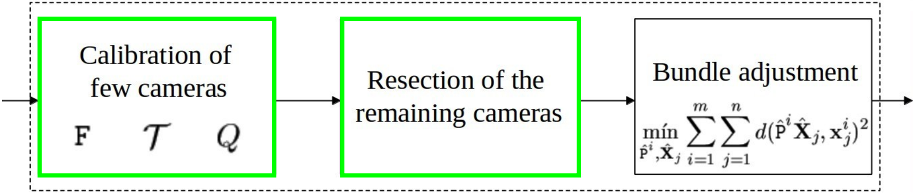
- We still need to provide an initialization.
- Use 2 views with many common features but sufficient baseline (parallax).



# SfM. Projective Calibration

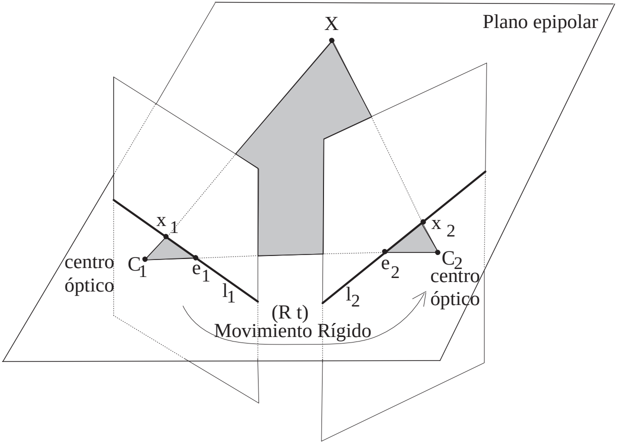
Recall where we are:

- We still need to provide an initialization.
- Use 2 views with many common features but sufficient baseline (parallax).



# Fundamental matrix F

- Useful for initializing an incremental reconstruction (3D model).
- F describes the relative geometry of two views of the same scene.



# Fundamental matrix F

## Problem statement

Given a set of $n$ point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ between two images, compute the fundamental matrix F satisfying the epipolar constraint

$$\mathbf{x}_i'^{\top} \mathtt{F} \mathbf{x}_i = 0 \qquad \text{for all} \quad i = 1, \ldots, n.$$

Estimation methods:

- Exact method ($n = 7$). Up to 3 solutions.
- Linear algorithm (algebraic cost, SVD, eigenvalues). $n \geq 8$
- Non-linear algorithm (geometric cost, Gauss-Newton method).
- Robust algorithm (RANSAC)

# Fundamental matrix $\mathtt{F}$

## Problem statement

Given a set of $n$ point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ between two images, compute the fundamental matrix $\mathtt{F}$ satisfying the epipolar constraint

$$\mathbf{x}_i'^{\top} \mathbf{F} \mathbf{x}_i = 0 \qquad \text{for all} \quad i = 1, \ldots, n.$$

Estimation methods:

- Exact method ($n = 7$). Up to 3 solutions.
- Linear algorithm (algebraic cost, SVD, eigenvalues). $n \geq 8$
- Non-linear algorithm (geometric cost, Gauss-Newton method).
- Robust algorithm (RANSAC)

- Each epipolar constraint is a linear equation in $\mathbf{f} \equiv \mathtt{F}$:

$$\mathbf{x}_i'^{\top} \mathtt{F} \mathbf{x}_i = \mathbf{a}_i^{\top} \mathbf{f} = 0.$$

- Check: $\mathbf{a}_i = (\mathbf{x}_i' \otimes \mathbf{x}_i)^{\top} \in \mathbb{R}^9$ if $\mathbf{f} = \mathrm{vec}(\mathtt{F}^{\top})$.

- Linear method: algebraic cost.
  - Singularity $\det(\mathtt{F}) = 0$ is enforced a posteriori.
  - Stack many equations: $\mathtt{A}\mathbf{f} = \mathbf{0}$, with $\mathtt{A} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n)^{\top} \in \mathbb{R}^{n \times 9}$.
  - Non-trivial, unique linear solution $\mathbf{f} = \ker(\mathtt{A})$ if $\mathrm{rank}(\mathtt{A}) = 8$.
  - Least-squares: minimize the Rayleigh quotient $\|\mathtt{A}\mathbf{f}\| / \|\mathbf{f}\|$, with

$$\|\mathtt{A}\mathbf{f}\|^2 = \sum_{i=1}^{n} (\mathbf{x}_i'^{\top} \mathtt{F} \mathbf{x}_i)^2.$$

- Each epipolar constraint is a linear equation in $\mathbf{f} \equiv \mathtt{F}$:

$$\mathbf{x}_i'^\top \mathtt{F} \mathbf{x}_i = \mathbf{a}_i^\top \mathbf{f} = 0.$$

- Check: $\mathbf{a}_i = (\mathbf{x}_i' \otimes \mathbf{x}_i)^\top \in \mathbb{R}^9$ if $\mathbf{f} = \operatorname{vec}(\mathtt{F}^\top)$.

- Linear method: algebraic cost.
  - Singularity $\det(\mathtt{F}) = 0$ is enforced a posteriori.
  - Stack many equations: $\mathtt{A}\mathbf{f} = \mathbf{0}$, with $\mathtt{A} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n)^\top \in \mathbb{R}^{n \times 9}$.
  - Non-trivial, unique linear solution $\mathbf{f} = \ker(\mathtt{A})$ if $\operatorname{rank}(\mathtt{A}) = 8$.
  - Least-squares: minimize the Rayleigh quotient $\|\mathtt{A}\mathbf{f}\| / \|\mathbf{f}\|$, with

$$\|\mathtt{A}\mathbf{f}\|^2 = \sum_{i=1}^{n} (\mathbf{x}_i'^\top \mathtt{F} \mathbf{x}_i)^2.$$

- Non-linear method: geometric cost function

$$g(\mathbf{f}, \{\mathbf{X}_i\}_{i=1}^{n}) = \sum_{i=1}^{n} \left( d^2_{\mathsf{Euc}}(\hat{\mathbf{x}}_i, \mathbf{x}_i) + d^2_{\mathsf{Euc}}(\hat{\mathbf{x}}'_i, \mathbf{x}'_i) \right).$$

where $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ truly satisfy $\hat{\mathbf{x}}'^{\top}_i \mathtt{F} \hat{\mathbf{x}}_i = 0$.

  - Singularity $\det(\mathtt{F}) = 0$ is enforced a priori (by parameterization).
  - New variables to be estimated: location of 3D points (that is why we use it to initialize the reconstruction).
  - Optimization in $12 + 3n$ parameters, but structure can be exploited to make it feasible and fast.
  - Solver: Levenberg-Marquardt algorithm (variant of Newton's method).

- RANSAC method for computing F is a standard (OpenCV, etc.)

- Non-linear method: geometric cost function

$$g(\mathbf{f}, \{\mathbf{X}_i\}_{i=1}^n) = \sum_{i=1}^{n} \left( d_{\mathsf{Euc}}^2(\hat{\mathbf{x}}_i, \mathbf{x}_i) + d_{\mathsf{Euc}}^2(\hat{\mathbf{x}}'_i, \mathbf{x}'_i) \right).$$

where $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ truly satisfy $\hat{\mathbf{x}}_i'^{\top} \mathtt{F} \hat{\mathbf{x}}_i = 0$.

  - Singularity $\det(\mathtt{F}) = 0$ is enforced a priori (by parameterization).
  - New variables to be estimated: location of 3D points (that is why we use it to initialize the reconstruction).
  - Optimization in $12 + 3n$ parameters, but structure can be exploited to make it feasible and fast.
  - Solver: Levenberg-Marquardt algorithm (variant of Newton's method).

- RANSAC method for computing F is a standard (OpenCV, etc.)

- Non-linear method: geometric cost function

$$g(\mathbf{f}, \{\mathbf{X}_i\}_{i=1}^{n}) = \sum_{i=1}^{n} \left( d^2_{\mathsf{Euc}}(\hat{\mathbf{x}}_i, \mathbf{x}_i) + d^2_{\mathsf{Euc}}(\hat{\mathbf{x}}'_i, \mathbf{x}'_i) \right).$$

  where $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ truly satisfy $\hat{\mathbf{x}}'^{\top}_i \mathtt{F} \hat{\mathbf{x}}_i = 0$.

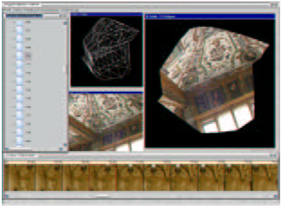
  - Singularity $\det(\mathtt{F}) = 0$ is enforced a priori (by parameterization).
  - New variables to be estimated: location of 3D points (that is why we use it to initialize the reconstruction).
  - Optimization in $12 + 3n$ parameters, but structure can be exploited to make it feasible and fast.
  - Solver: Levenberg-Marquardt algorithm (variant of Newton's method).
- RANSAC method for computing F is a standard (OpenCV, etc.)

# Outline

# VISIRE

## Virtual Image Processing System for Intelligent Reconstruction of 3D Environments.

- European project IST-1999-10756. 5th Framework Programme.
- Goal: semi-automatic photorealistic 3D reconstructions from video sequences.

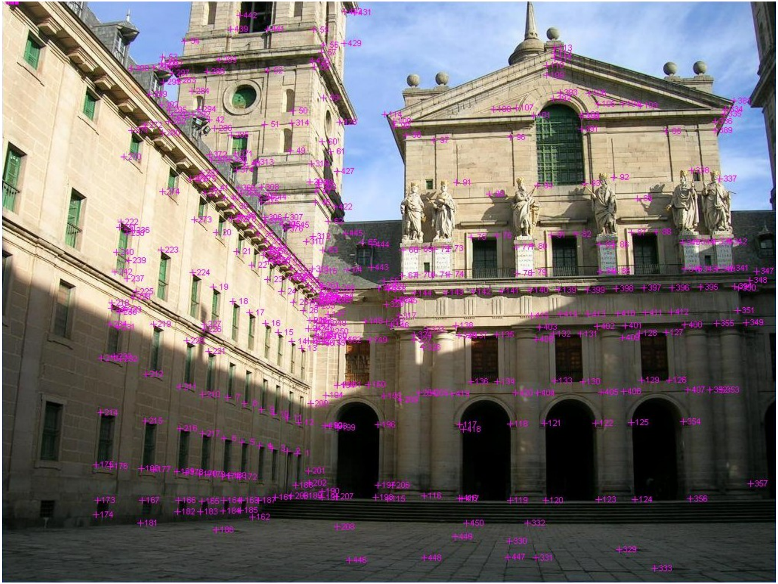


# Outline

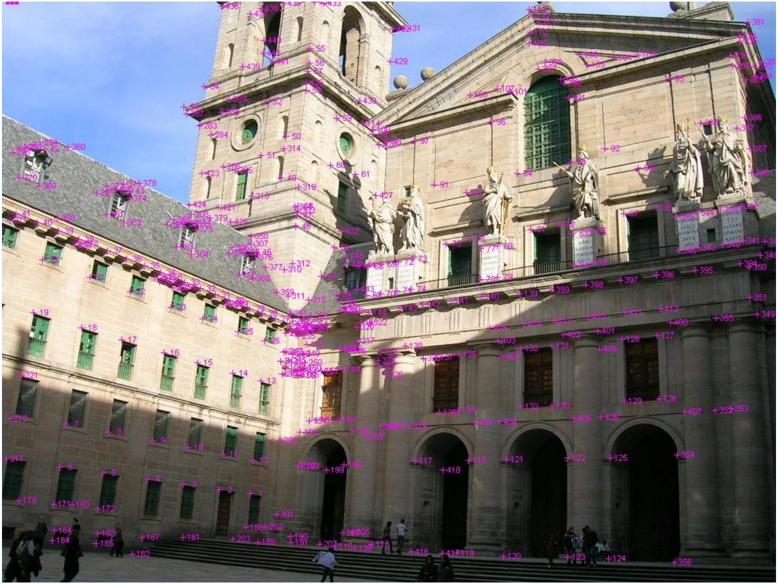
# Recall the Processing Steps

1. Digital image acquisition
2. Feature detection and extraction
3. Structure from Motion
    1. Projective calibration
    2. Auto-calibration
    3. Euclidean reconstruction
4. 3D triangulation and texture selection
5. VRML presentation

# Feature (corner) extraction. Image 1



# Feature (corner) extraction. Image 23



# SfM. Projective Calibration

>> calib_tool3
Recognizing 450 points in 23 images.
Normalization done
Gold Standard Algorithms for Projective Calibration
Residual Reprojection **error = 1.301 pixels**

Optimal Projective Calibration (Bundle Adjustment)
Previous calibration detected Residual Reprojection **error = 0.586 pixels**

Mean number of points per image = 372.65
Estimated noise level (sigma) = 0.61 pixels
Residual Reprojection error (projective reconstruction) = 0.586 pixels
Pixel error: mean = [ -0.00355 0.00025]
Pixel error: std = [ 0.71089 0.42730]

# SfM. Projective Calibration

>> calib_tool3
Recognizing 450 points in 23 images.
Normalization done
Gold Standard Algorithms for Projective Calibration
Residual Reprojection **error = 1.301 pixels**

Optimal Projective Calibration (Bundle Adjustment)
Previous calibration detected Residual Reprojection **error = 0.586 pixels**

Mean number of points per image = 372.65
Estimated noise level (sigma) = 0.61 pixels
Residual Reprojection error (projective reconstruction) = 0.586 pixels
Pixel error: mean = [ -0.00355 0.00025]
Pixel error: std = [ 0.71089 0.42730]

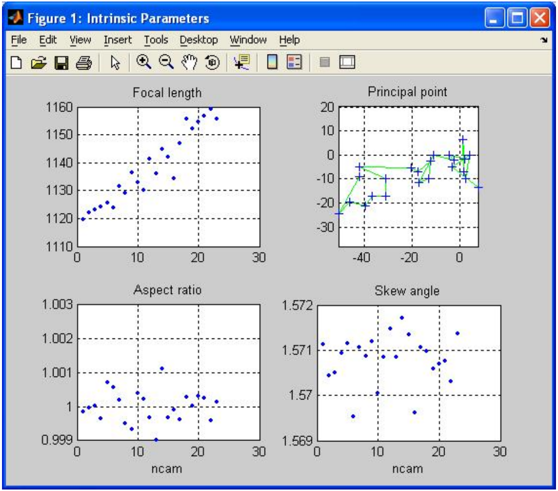
# SfM. Projective Calibration

```
>> calib_tool3
```

Recognizing 450 points in 23 images.
Normalization done
Gold Standard Algorithms for Projective Calibration
Residual Reprojection **error = 1.301 pixels**

Optimal Projective Calibration (Bundle Adjustment)
Previous calibration detected Residual Reprojection **error = 0.586 pixels**

Mean number of points per image = 372.65
Estimated noise level (sigma) = 0.61 pixels
Residual Reprojection error (projective reconstruction) = 0.586 pixels
Pixel error: mean = [ -0.00355 0.00025]
Pixel error: std = [ 0.71089 0.42730]



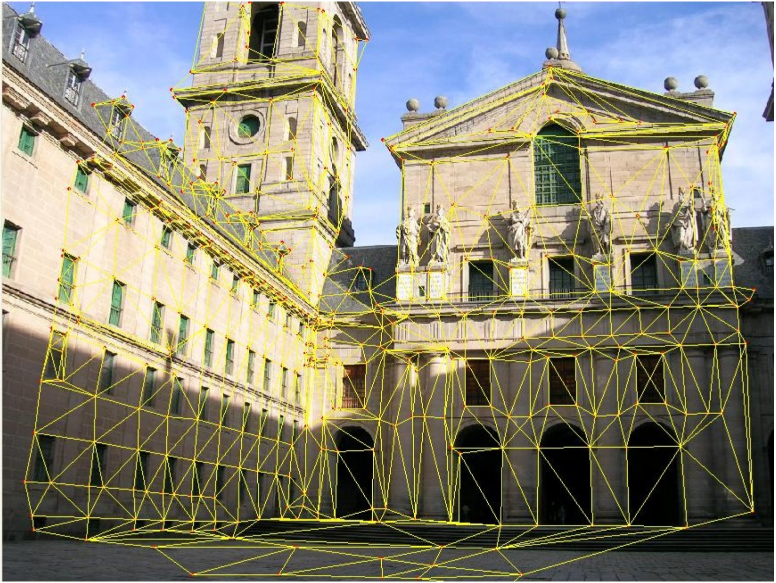

Algorithm: minimization of the error in the pixel shape

# SfM. Euclidean Bundle Adjustment

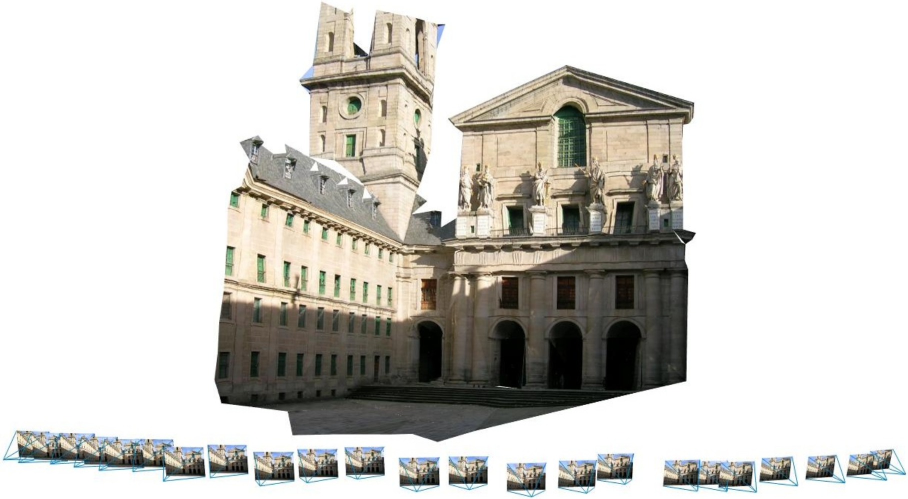
- Euclidean Bundle Adjustment
  Residual Reprojection **error = 0.592 pixels**

- It is slightly larger than the reprojection error of the projective reconstruction because the Euclidean reconstruction has less degrees of freedom (less parameters to decrease the error).

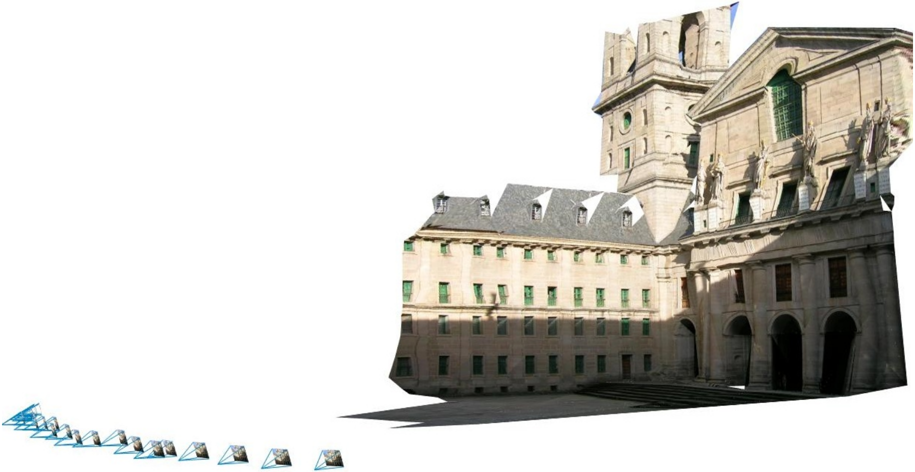
# 3D Mesh generation and Texture selection



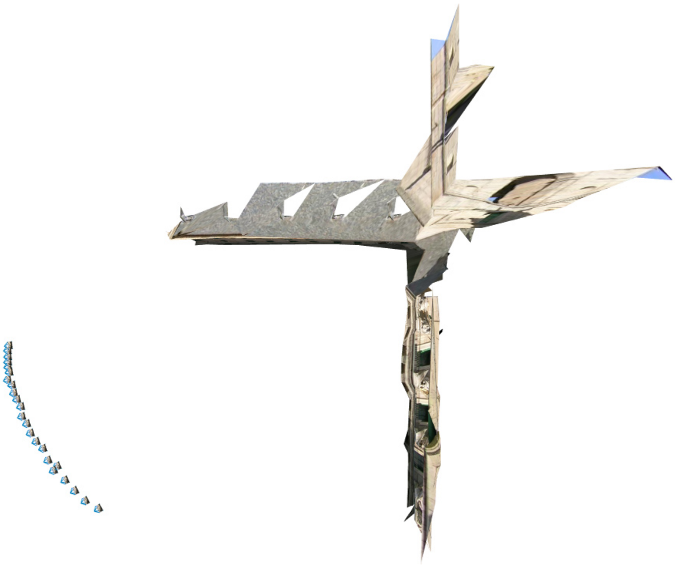
# VRML model: Escorial



# VRML model: Escorial (side view)



# VRML model: Escorial (top view)

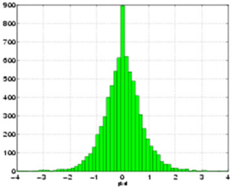


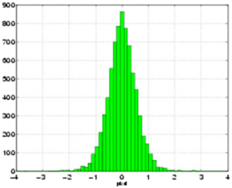
# Image error summary

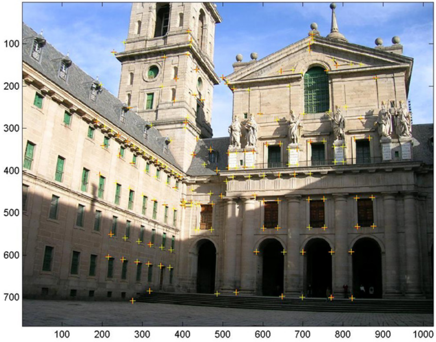
| Etapa | Error reproyección RMS (píxeles) |
|---|---|
| Calibración proyectiva inicial | 0,7399 |
| Tras el ajuste de haces proyectivo | 0,5508 |
| Calibración euclídea inicial (píxeles cuadrados) | 1,5878 |
| Tras el ajuste de haces euclídeo | 0,5528 |

$$f : \mathbb{R}^{762} \longrightarrow \mathbb{R}^{7452}$$

$$\mathtt{K}^{12} = \begin{bmatrix} 1157{,}9 & 0 & 21{,}4 \\ 0 & 1157{,}9 & -13{,}3 \\ 0 & 0 & 1 \end{bmatrix}$$



# Outline

# VRML model: Books

# VRML model: Books

# VRML model: Books

# VRML model: Books

# VRML model: Books

# Outline

# Structure from Motion. Photo Tourism

(Snavely, Seitz, Szeliski) `http://phototour.cs.washington.edu/applet/index.html`


(a) (b) (c)

A system for browsing large collections of photographs in 3D.

1. Input: large collections of images (a)
2. Process: automatically compute each photo's viewpoint and a sparse 3D model of the scene.
3. Display & interactively move about the 3D space.

VERY popular because the SfM code is available online: **Bundler**.

# Structure from Motion. Photo Tourism

(Snavely, Seitz, Szeliski) `http://phototour.cs.washington.edu/applet/index.html`


(a) (b) (c)

A system for browsing large collections of photographs in 3D.

1. Input: large collections of images (a)
2. Process: automatically compute each photo's viewpoint and a sparse 3D model of the scene.
3. Display & interactively move about the 3D space.

VERY popular because the SfM code is available online: **Bundler**.

# Structure from Motion. Photo Tourism

(Snavely, Seitz, Szeliski) `http://phototour.cs.washington.edu/applet/index.html`

(a) (b) (c)

A system for browsing large collections of photographs in 3D.

1. Input: large collections of images (a)
2. Process: automatically compute each photo's viewpoint and a sparse 3D model of the scene.
3. Display & interactively move about the 3D space.

VERY popular because the SfM code is available online: **Bundler**.

# PhotoSynth (Microsoft & U. Washington).

http://photosynth.net/

Photosynth is based on Photo Tourism. It is software for:

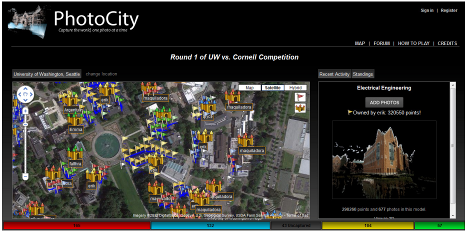
- 3D modeling (location of the cameras in the scene & sparse point cloud)
- Generation of huge panoramas (100's Mpixels), since 2010.

More user-friendly than Bundler (research code), but less configurable.

# PhotoCity Game (U.Washington & Cornell)

http://grail.cs.washington.edu/projects/photocity/

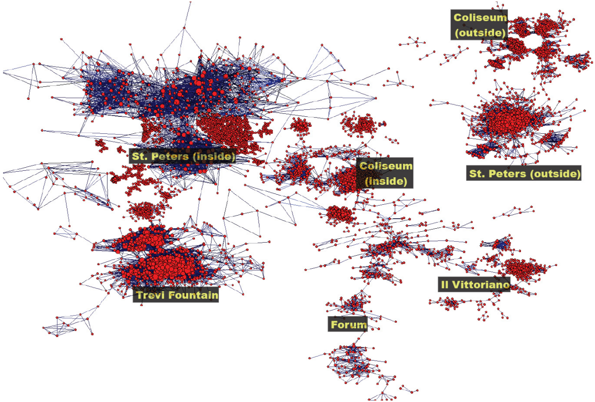
- PhotoCity is a game for reconstructing the world in 3D out of photos.
- Players take pictures of building exteriors from all different angles, which are then used to *automatically generate 3D models* and calculate virtual ownership of the buildings.
- Our ultimate goal is to reconstruct the entire world, one photo at a time.




# City-scale SfM

Building Rome in a day (Agarwal et al, ICCV 2009)

- ∼500 cores, ∼200.000 images, 1 day of processing (from image matching to large scale optimization).
- Experiment in 3 cities: Rome, Venice, Dubrovnik. `http://www.youtube.com/watch?v=kxtQqYLRaSQ`



# City-scale SfM

Building Rome in a day (Agarwal et al, ICCV 2009)

- ~500 cores, ~200.000 images, 1 day of processing (from image matching to large scale optimization).
- Experiment in 3 cities: Rome, Venice, Dubrovnik.
  `http://www.youtube.com/watch?v=kxtQqYLRaSQ`

Colosseum: 2,097 images, 819,242 points    Trevi Fountain: 1,935 images, 1,055,153 points

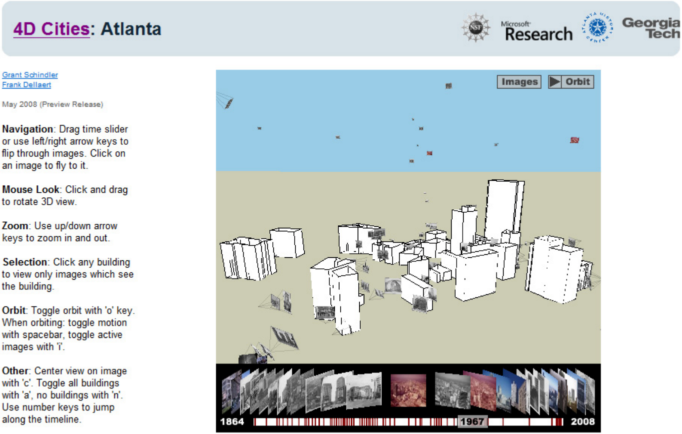
Pantheon: 1,032 images, 530,076 points    Hall of Maps: 275 images, 230,182 points

(b) Rome: Four of the largest connected components visualized at canonical viewpoints [14].

San Marco Square: 13,699 images, 4,515,157 points

4D Cities (Georgia Tech, 2007)

- Browse large collection of photos in space and time. Show history of the city.
- Space-time reconstruction (based on Bundler) and navigation.
- Model of Atlanta: `http://4d-cities.cc.gatech.edu/atlanta/`



# Outline

1. Introduction
   - Motivation
   - Classification of 3D reconstruction methods
2. Image based 3D reconstruction
   - Processing steps
   - SfM. Problem Statement
   - Projective Calibration
   - SfM. Camera resectioning
   - SfM. Initial camera pair estimation
3. Examples
   - VISIRE project (2000)
   - ADREP 3D project (2003). San Lorenzo de El Escorial Monastery
   - ADREP 3D project (2003). Books scene
   - Photo Tourism (2006), PhotoSynth (2008), etc.
   - Multi-View Stereo: PMVS (2007) & CMVS (2010)
   - Autocalibration. Six-Line Conic Variety (SLCV)
   - Video Surfing project. Telefonica R&D (2009)

# Multi-View Stereo: PMVS & CMVS
## (Furukawa and Ponce).

Patch-based Multi-View Stereo (PMVS). CVPR 2007.

- Computes a *dense* reconstruction of the scene using patches (vs points).
- **Cameras are calibrated**. The only unknown is the structure of the scene.
- Again, very popular because the code is available online.
- http://grail.cs.washington.edu/software/pmvs/gallery.html

Clustering Views for Multi-view Stereo (CMVS). CVPR 2010.

- Management of a very large data set (divide-and-conquer):
  - Split the dataset into clusters of images (using graph partitioning methods) to compute partial 3D reconstructions.
  - Merge the 3D reconstructions.
- http://www.youtube.com/watch?v=ofHFOr2nRxU

# Multi-View Stereo: PMVS & CMVS

(Furukawa and Ponce).

Patch-based Multi-View Stereo (PMVS). CVPR 2007.

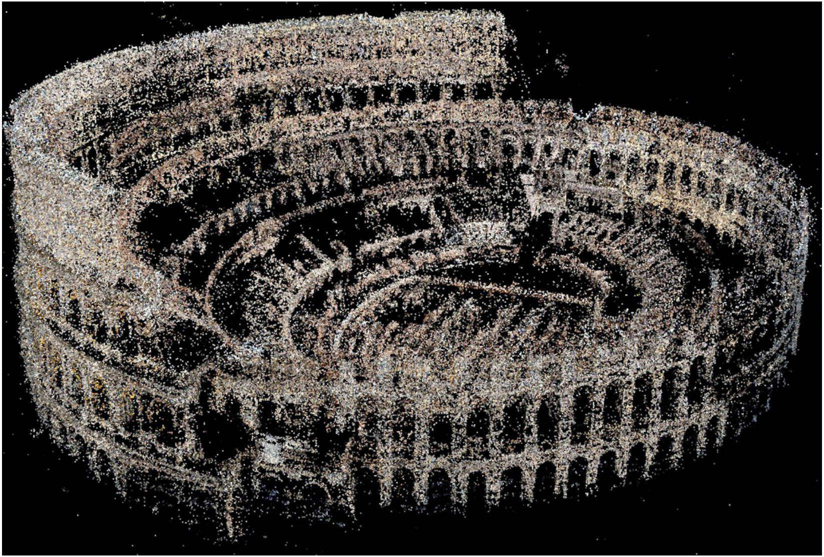
- Computes a *dense* reconstruction of the scene using patches (vs points).
- **Cameras are calibrated**. The only unknown is the structure of the scene.
- Again, very popular because the code is available online.
- http://grail.cs.washington.edu/software/pmvs/gallery.html

Clustering Views for Multi-view Stereo (CMVS). CVPR 2010.

- Management of a very large data set (divide-and-conquer):
  - Split the dataset into clusters of images (using graph partitioning methods) to compute partial 3D reconstructions.
  - Merge the 3D reconstructions.
- http://www.youtube.com/watch?v=ofHFOr2nRxU

# Sparse vs. Dense reconstruction

## Sparse: output of Structure from Motion (SfM: Bundler)



# Sparse vs. Dense reconstruction

## Dense: SfM $\longrightarrow$PMVS (output)

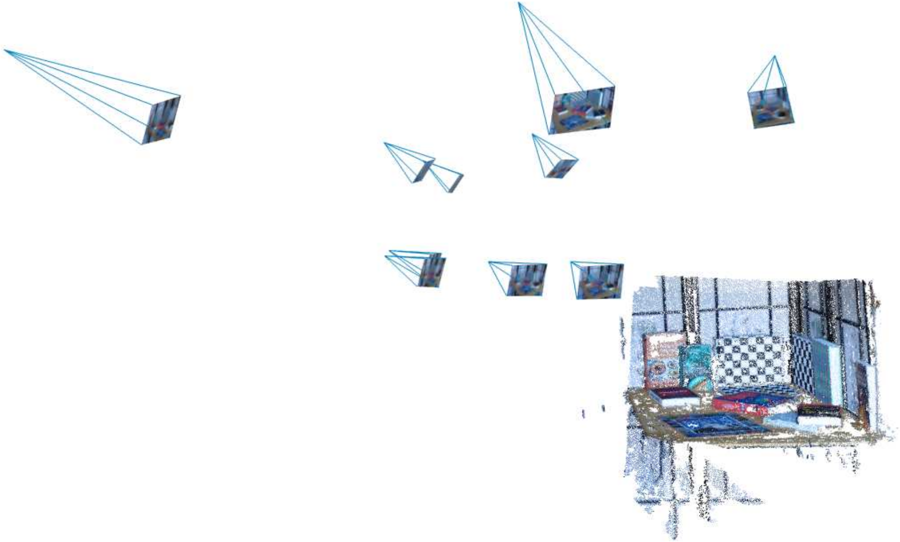
# Outline

# Books (SLCV)

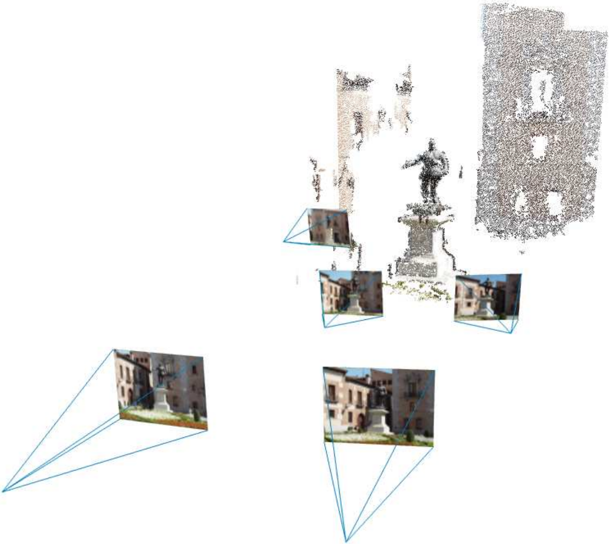
Using PMVS for visualizing the result (dense reconstruction) of autocalibration methods.

# Books (SLCV)

Using PMVS for visualizing the result (dense reconstruction) of autocalibration methods.



# Plaza De La Villa (Madrid)

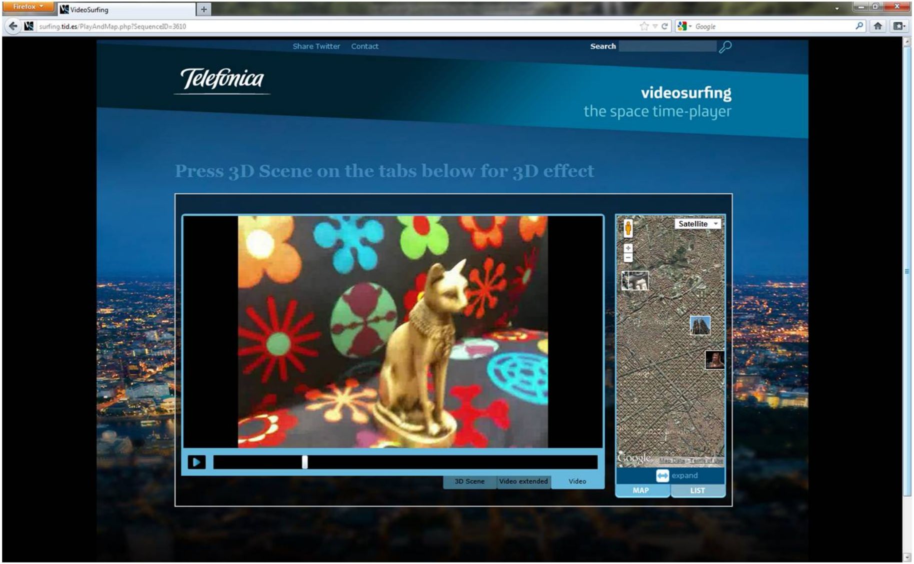


# Plaza De La Villa (Madrid)

# Outline

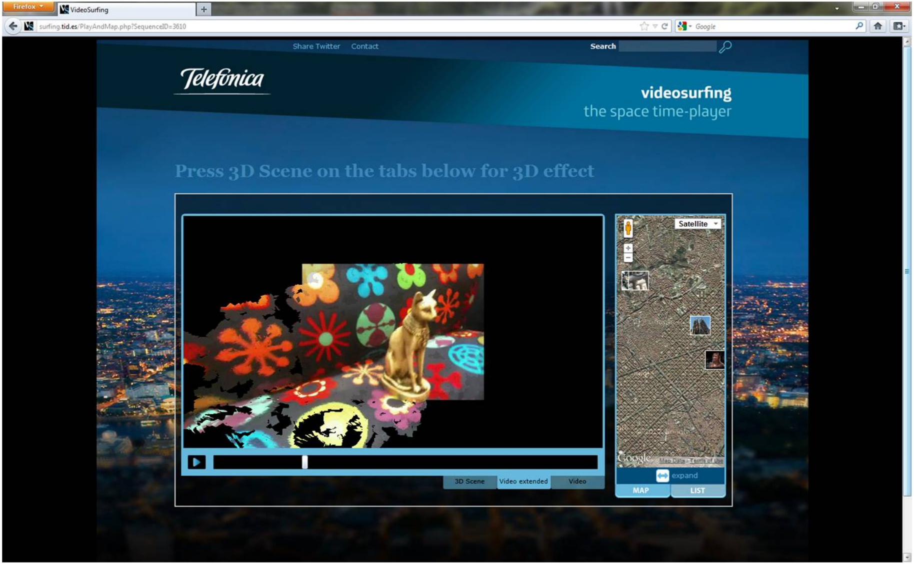
# Video Surfing. http://vimeo.com/15990190

3 modes of interaction: video player.



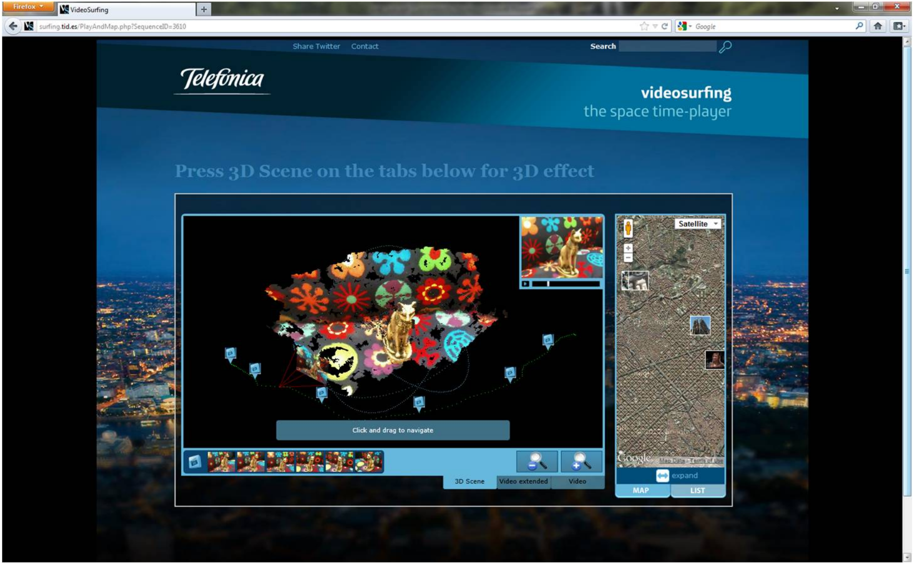
# Video Surfing. http://vimeo.com/15990190

3 modes of interaction: extended video player (behind the camera).



# Video Surfing. http://vimeo.com/15990190

3 modes of interaction: free-flight navigation in the 3D scene.



# Summary

1. Advances in technology arise from connections between geometry, algebra, optimization, statistics, etc.
2. Improve your habilities / toolbox in the above areas.
3. Overview of recent results in (image based) multi-view stereo (3D) reconstruction.

# Summary

1. Advances in technology arise from connections between geometry, algebra, optimization, statistics, etc.
2. Improve your habilities / toolbox in the above areas.
3. Overview of recent results in (image based) multi-view stereo (3D) reconstruction.

# Summary

1. Advances in technology arise from connections between geometry, algebra, optimization, statistics, etc.
2. Improve your habilities / toolbox in the above areas.
3. Overview of recent results in (image based) multi-view stereo (3D) reconstruction.

# References I

Richard Hartley and Andrew Zissermann.
*Multiple View Geometry in Computer Vision (2nd Ed.)*
Cambridge University Press, March 2004.

Guillermo Gallego.
*Sistema de Autocalibración de Cámaras y Reconstrucción 3D*.
*Proyecto Fin de Carrera*, ETSI Telecomunicación, February 2004.
Tutor: José Ignacio Ronda Prieto.